

Descrição da Task para Candidato à Vaga de Estágio Back-End

Objetivo: Desenvolver uma aplicação back-end simples que inclua as seguintes funcionalidades:

1. Autenticação de usuários
2. CRUD de vendas (consultar, adicionar, editar e excluir vendas)
3. Gerar um PDF contendo as vendas em um intervalo de datas

Requisitos Técnicos:

- Utilização de Python com Flask ou Node.js com Express
- Banco de dados SQLite
- Implementação de autenticação utilizando JWT
- Arquitetura REST

Instruções Detalhadas:

1. Autenticação de Usuários:

- **Funcionalidade:** Permitir que o usuário faça login e registro.
- **Endpoints:**
 - **POST /register:** Registro de novos usuários.
 - **Campos:** email, senha
 - **POST /login:** Autenticação de usuários existentes.
 - **Campos:** email, senha
- **Autenticação:**
 - Implementar autenticação utilizando JWT
 - Proteger os endpoints de vendas com autenticação JWT

2. CRUD de Vendas:

- **Funcionalidade:** Permitir ao usuário consultar, adicionar, editar e excluir vendas.
- **Endpoints:**
 - **GET /sales:** Consultar todas as vendas.
 - **POST /sales:** Adicionar uma nova venda.
 - **Campos:** nome_cliente, produto, valor, data_venda
 - **PUT /sales/:id:** Editar uma venda existente.
 - **Campos:** nome_cliente, produto, valor, data_venda
 - **DELETE /sales/:id:** Excluir uma venda existente.

3. Gerar PDF das Vendas:

- **Funcionalidade:** Permitir ao usuário gerar um PDF contendo todas as vendas em um intervalo de datas. Cada venda no PDF deve mostrar informações básicas como ID da venda, nome do cliente, produto, valor e data da venda.
- **Endpoints:**

- GET /sales/pdf?start_date=DD-MM-YYYY&end_date=DD-MM-YYYY:
Gerar um PDF contendo todas as vendas entre `start_date` e `end_date`.

Detalhes do Banco de Dados:

- Tabelas:
 - Usuários:
 - `id` (inteiro, chave primária)
 - `email` (texto, único)
 - `senha` (texto)
 - Vendas:
 - `id` (inteiro, chave primária)
 - `nome_cliente` (texto)
 - `produto` (texto)
 - `valor` (decimal)
 - `data_venda` (data)
 - `user_id` (inteiro, chave estrangeira para `Usuários`)

Entrega:

- **Repositório Git:** O candidato deve criar um repositório Git público e enviar o link com a aplicação desenvolvida.
- **Documentação:**
 - README.md explicando como configurar e executar a aplicação.
 - Breve descrição das escolhas tecnológicas e arquitetônicas.

Critérios de Avaliação:

- Funcionalidade: Todas as funcionalidades descritas estão implementadas corretamente.
- Qualidade do Código: Código limpo, bem estruturado e comentado.
- Segurança: Implementação correta da autenticação JWT.
- Documentação das funcionalidades

Prazo: O candidato terá 1 semana para completar e enviar a task. Se houver qualquer dúvida ou problema durante o desenvolvimento, pode entrar em contato com a equipe pelo email: suporte@solucaotributos.com.br.

Boa sorte! Estamos ansiosos para ver o seu trabalho!